

# Notes on Rigid Body Transformation

Gyubeom Edward Im\*

November 30, 2023

## Contents

<b>1 Rigid body transformation</b>	<b>2</b>
<b>2 Coordinate system</b>	<b>2</b>
<b>3 Rotation matrix</b>	<b>3</b>
3.1 Properties of rotation matrix . . . . .	4
3.2 SO(3) group . . . . .	6
<b>4 Transformation matrix</b>	<b>6</b>
4.1 Properties of transformation matrix . . . . .	7
4.2 SE(3) group . . . . .	10
<b>5 Other rotation representations</b>	<b>10</b>
5.1 Euler angle . . . . .	10
5.1.1 Various euler angle representations . . . . .	10
5.1.2 Relationship between euler angle and rotation matrix . . . . .	11
5.1.3 Limitation of euler angle . . . . .	12
5.2 Fixed angle (roll-pitch-yaw) . . . . .	12
5.3 Angle-axis representation . . . . .	13
5.4 Properties of angle-axis representation . . . . .	13
5.5 Quaternion . . . . .	14
5.5.1 Properties of quaternion . . . . .	15
5.5.2 Unit quaternion as a rotation operator . . . . .	15
5.5.3 Interpolation . . . . .	16
<b>6 Relationship between rotation representations</b>	<b>16</b>
6.1 Angle-axis and quaternion . . . . .	16
6.2 Rotation matrix and quaternion . . . . .	17
6.3 Rotation matrix and angle-axis . . . . .	17
6.4 Quaternion and euler angle(ZYX) . . . . .	17
6.5 Rotation matrix and euler angle(ZYX) . . . . .	18
<b>7 Reference</b>	<b>18</b>
<b>8 Revision log</b>	<b>18</b>
<b>A Various transformations</b>	<b>18</b>
A.1 Isometry transformation . . . . .	19
A.2 Similarity transformation . . . . .	19
A.3 Affine transformation . . . . .	19
A.4 Projective transformation . . . . .	19

---

\*blog: alida.tistory.com, email: criterion.im@gmail.com

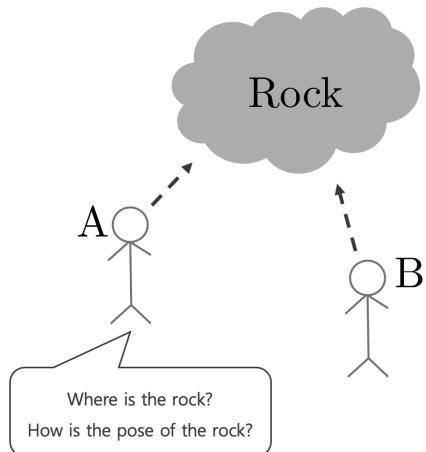
# 1 Rigid body transformation

본 자료에서는 3차원 공간 상의 강체(rigid body)의 움직임을 수학적으로 표현하는 다양한 방법들에 대하여 설명한다. 강체란 이론적으로 무한대의 강성을 가지는 물체를 뜻한다. 따라서 강체는 외부에서 힘을 가해도 모양이 변형이 되지 않으며 무게중심으로 지정되는 하나의 점에 의해 운동이 결정된다. 강체 변환 이외에도 물체를 변환시키는 다양한 연산이 존재하는데 자세한 내용은 Appendix A를 참조하면 된다.

SLAM에서 강체란 일반적으로 움직이는 카메라 센서를 의미하며 시간에 따라 카메라의 형태가 일정하다고 가정하므로 강체 변환(rigid body transformation)을 사용하여 카메라의 움직임을 수학적으로 표현할 수 있다.

## 2 Coordinate system

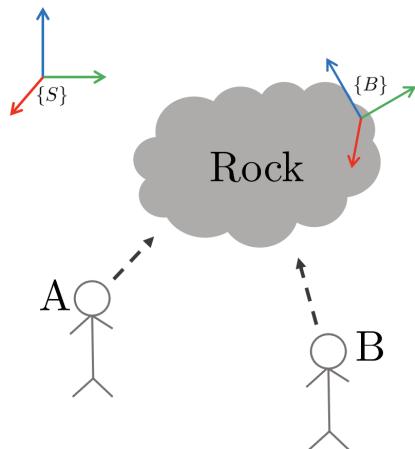
본 섹션에서는 이후 설명할 모든 변환의 기준이 되는 좌표계(coordinate system)에 대해 설명한다. 좌표계는 물체의 위치 및 방향을 수학적으로 표현할 수 있게 해주는 중요한 도구이다. 좌표계에 대해 자세히 이해하기 위해 아래 예시를 살펴 보자. 본 예시는 Keunjun님 블로그 포스팅의 내용을 주로 참고하였다.



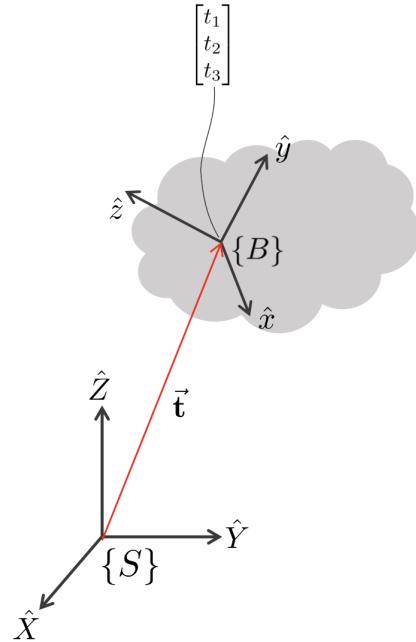
예를 들어 두 사람이 하나의 돌을 바라보고 있을 때 돌을 표현하는 방법이 서로 다를 수 있다. A가 B에게 "돌이 어디에 어떤 자세로 있어?"라고 물었을 때 B는 어떻게 대답해야 할까? 정답은 알 수 없다. 엄밀하게 말하면 질문에 오류가 있다.

- 어디를 기준으로 돌이 어디에 있는가?
- 돌의 표면 중 정확히 어느 점을 물어보고 있는가?

이처럼 물체의 위치와 자세를 이야기할 때 어디를 기준으로 표현하는가?(기준좌표계  $\{S\}$ ) 그리고 그 물체의 어디를 말하고자 하는가?(이동좌표계  $\{B\}$ )를 명확하게 해야 한다.



일반적으로 기준이 되는 좌표계  $\{S\}$ 를 기준좌표계(fixed frame, world frame, space frame)으로 부른다. 그리고 이동하는 물체를 대표하는 좌표계  $\{B\}$ 를 이동좌표계(moving frame, body frame)으로 부른다.



강체의 경우  $\{B\}$  좌표계를 아무 곳이나 물체의 한 곳에 잡으면 된다. 보통은 무게중심에 잡는다.  $\{B\}$  좌표계가 움직여도 강체는 형태가 변하지 않으므로 강체의 다른 부분의 위치가 자동으로 결정된다. **이동좌표계  $\{B\}$  의 원점을  $\vec{t}$ , 축을  $(\hat{x}, \hat{y}, \hat{z})$ 라고 하고 기준좌표계  $\{S\}$  의 축을  $(\hat{X}, \hat{Y}, \hat{Z})$ 라고 하자.** 이 때 기준좌표계  $\{S\}$ 를 기준으로 이동좌표계  $\{B\}$ 의 원점을 표현해보면 다음과 같다.

$$\vec{t} = t_1 \hat{X} + t_2 \hat{Y} + t_3 \hat{Z} \quad \cdots \{B\}'s \text{ origin} \quad (1)$$

다음으로 이동좌표계  $\{B\}$ 의 축  $(\hat{x}, \hat{y}, \hat{z})$ 를  $(\hat{X}, \hat{Y}, \hat{Z})$ 의 선형결합으로 표현하면 다음과 같다.

$$\begin{aligned} \hat{x} &= r_{11} \hat{X} + r_{21} \hat{Y} + r_{31} \hat{Z} \\ \hat{y} &= r_{12} \hat{X} + r_{22} \hat{Y} + r_{32} \hat{Z} \quad \cdots \{B\}'s \text{ axis} \\ \hat{z} &= r_{13} \hat{X} + r_{23} \hat{Y} + r_{33} \hat{Z} \end{aligned} \quad (2)$$

### 3 Rotation matrix

앞서 설명한  $\{B\}$  좌표계의 표현 (1), (2)를 합쳐서 표시하면 다음과 같다.

$$\begin{aligned} \vec{t} &= t_1 \hat{X} + t_2 \hat{Y} + t_3 \hat{Z} \\ \hat{x} &= r_{11} \hat{X} + r_{21} \hat{Y} + r_{31} \hat{Z} \\ \hat{y} &= r_{12} \hat{X} + r_{22} \hat{Y} + r_{32} \hat{Z} \\ \hat{z} &= r_{13} \hat{X} + r_{23} \hat{Y} + r_{33} \hat{Z} \end{aligned} \quad (3)$$

위 식을 벡터와 행렬 형태로 나타내면 다음과 같이 간결하게 나타낼 수 있다.

$$\vec{t} = \mathbf{t}^T \begin{bmatrix} \hat{X} \\ \hat{Y} \\ \hat{Z} \end{bmatrix} \quad \begin{bmatrix} \hat{x} \\ \hat{y} \\ \hat{z} \end{bmatrix} = \mathbf{R} \begin{bmatrix} \hat{X} \\ \hat{Y} \\ \hat{Z} \end{bmatrix} \quad (4)$$

이 때,  $(\hat{X}, \hat{Y}, \hat{Z}) \rightarrow (\hat{x}, \hat{y}, \hat{z})$  변환은 크기는 변하지 않고 오직 회전에만 관여하는데 이러한 변환에

사용하는 행렬  $\mathbf{R}$  을 회전 행렬(rotation matrix) 이라고 한다.

$$\boxed{\mathbf{R} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \in \mathbb{R}^{3 \times 3}} \quad (5)$$

$\mathbf{R}$  은 보통  ${}_B^S \mathbf{R}$  또는  $\mathbf{R}_{SB}$  로 표기한다. 이 표현의 의미는 다음과 같이 두 가지로 해석할 수 있다.

- 행렬 자체로 봤을 때:  $\{S\}$  좌표계에서 바라본  $\{B\}$  좌표계의 회전(량) 또는 방향 (orientation of  $\{B\}$  frame with respect to  $\{S\}$  frame)
- 연산자로 봤을 때: 점 또는 벡터를  $\{B\} \rightarrow \{S\}$  좌표계로 변환하는 연산자

이는 다음 섹션에서 자세하게 설명한다.

### 3.1 Properties of rotation matrix

이전 섹션에서 본 것과 같이  $\{B\}$  좌표계를 표현하려면 다음과 같이 12개의 파라미터가 필요하다.

$$(t_1, t_2, t_3), (r_{11}, r_{12}, r_{13}, r_{21}, r_{22}, r_{23}, r_{31}, r_{32}, r_{33}) \quad (6)$$

하지만 3차원 공간에서 물체는 회전 3자유도(roll, pitch, yaw)와 이동 3자유도(x,y,z)를 합한 6자유도(=6dof, 6 degree of freedom)를 갖는다. 이에 따라 12개의 변수는 모두 독립이 아니고 종속적이라는 것을 알 수 있다. 즉, 회전 행렬에서 3개의 값을 독립적으로 정하면 제약조건에 의해 나머지 6개의 변수는 결정된다는 것을 알 수 있다. 회전 행렬의 제약 조건은 다음과 같다.

1.  $\{B\}$  좌표계의 축들은 단위벡터라는 조건

$$\begin{aligned} (\text{i}) \quad |\hat{x}|^2 &= 1 : r_{11}^2 + r_{21}^2 + r_{31}^2 = 1 \\ (\text{ii}) \quad |\hat{y}|^2 &= 1 : r_{12}^2 + r_{22}^2 + r_{32}^2 = 1 \\ (\text{iii}) \quad |\hat{z}|^2 &= 1 : r_{13}^2 + r_{23}^2 + r_{33}^2 = 1 \end{aligned} \quad (7)$$

2.  $\{B\}$  좌표계의 축들은 서로 수직하다는 조건

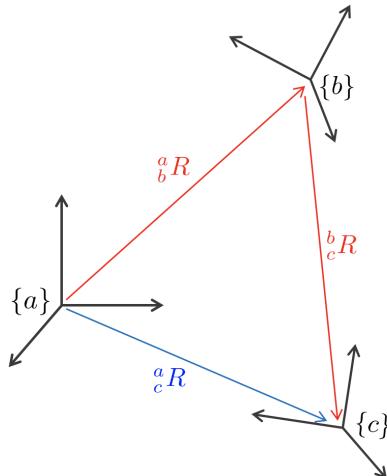
$$\begin{aligned} (\text{iv}) \quad \hat{x} \perp \hat{y} : r_{11}r_{12} + r_{21}r_{22} + r_{31}r_{32} &= 0 \\ (\text{v}) \quad \hat{x} \perp \hat{z} : r_{11}r_{13} + r_{21}r_{23} + r_{31}r_{33} &= 0 \\ (\text{vi}) \quad \hat{y} \perp \hat{z} : r_{12}r_{13} + r_{22}r_{23} + r_{32}r_{33} &= 0 \end{aligned} \quad (8)$$

- 위 조건들을 합쳐서 보통 다음과 같이 표현한다. (i ~ vi)  $\mathbf{R}^T \mathbf{R} = \mathbf{I}$ .
- 이를 회전 행렬의 직교 성질(orthogonal property)라고 한다.

3. (일반적으로) 오른손 좌표계를 사용한다는 조건

$$(\text{vii}) \quad \det \mathbf{R} = 1 \Leftrightarrow \hat{x} \times \hat{y} = \hat{z} \quad (9)$$

4. 다음과 같은 연속적인 회전을 나타낼 때는 회전 행렬의 곱셈을 통해 표현할 수 있다.

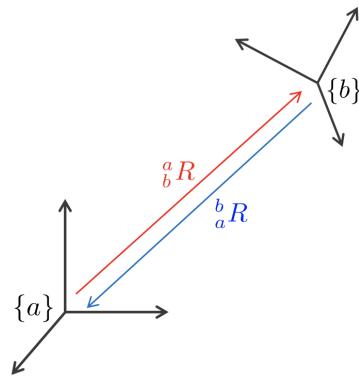


\*직관적인 설명을 위해 좌표계가 이동한 것처럼 표시했다.  
실제로는 원점이 같은 상태에서 오직 회전만 고려한다.

$${}_b^a \mathbf{R} {}_c^b \mathbf{R} = {}_c^a \mathbf{R} \quad (10)$$

$$\begin{aligned} [\hat{x}_b, \hat{y}_b, \hat{z}_b] &= [\hat{x}_a, \hat{y}_a, \hat{z}_a] {}_b^a R \\ [\hat{x}_c, \hat{y}_c, \hat{z}_c] &= [\hat{x}_b, \hat{y}_b, \hat{z}_b] {}_c^b R \\ \therefore [\hat{x}_c, \hat{y}_c, \hat{z}_c] &= [\hat{x}_a, \hat{y}_a, \hat{z}_a] {}_c^a R {}_b^b R \end{aligned} \quad (11)$$

5.  $\{a\} \rightarrow \{b\}$  만큼 회전한 다음 다시  $\{b\} \rightarrow \{a\}$  만큼 회전하면 원래  $\{a\}$  좌표계가 된다. 즉, 회전하지 않은 것과 동일하다.



\*직관적인 설명을 위해 좌표계가 이동한 것처럼 표시했다.  
실제로는 원점이 같은 상태에서 오직 회전만 고려한다.

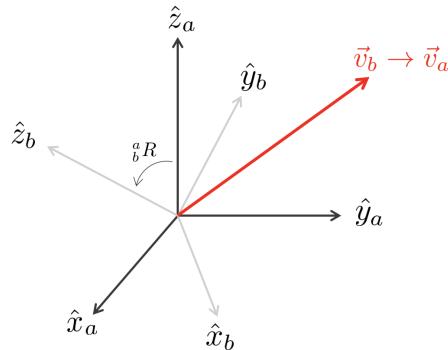
$${}_b^a \mathbf{R} {}_a^b \mathbf{R} = \mathbf{I} \quad (12)$$

$${}_a^a \mathbf{R} {}_b^a \mathbf{R}^T = \mathbf{I} \quad \text{and} \quad {}_b^a \mathbf{R} {}_a^b \mathbf{R} = \mathbf{I} \quad (13)$$

6. 위 식으로부터 회전 행렬은 역행렬과 전치행렬이 동일 하며 이는 반대방향으로 회전을 나타냄을 알 수 있다.

$$\begin{aligned} \mathbf{R}^T \mathbf{R} = \mathbf{I} &\leftrightarrow \mathbf{R}^T = \mathbf{R}^{-1} \\ \therefore {}_b^a \mathbf{R}^{-1} &= {}_b^a \mathbf{R}^T = {}_a^b \mathbf{R} \end{aligned} \quad (14)$$

7. 회전 행렬을 통해 특정 벡터의 기준 좌표계 또한 변환할 수 있다.



- $\vec{v}$  벡터를  $\{a\}$  좌표계와  $\{b\}$  좌표계에서 각각 표현하면 다음과 같다.

$$\vec{v}_a = v_{ax}\hat{x}_a + v_{ay}\hat{y}_a + v_{az}\hat{z}_a = \begin{bmatrix} \hat{x}_a & \hat{y}_a & \hat{z}_a \end{bmatrix} \begin{bmatrix} v_{ax} \\ v_{ay} \\ v_{az} \end{bmatrix}$$

$$\vec{v}_b = v_{bx}\hat{x}_b + v_{by}\hat{y}_b + v_{bz}\hat{z}_b = \begin{bmatrix} \hat{x}_b & \hat{y}_b & \hat{z}_b \end{bmatrix} \begin{bmatrix} v_{bx} \\ v_{by} \\ v_{bz} \end{bmatrix}$$

- 이 때  $\{b\}$  좌표계는  $\{a\}$  좌표계와 회전 행렬을 통해 표현이 가능하므로

$$\begin{bmatrix} \hat{x}_b & \hat{y}_b & \hat{z}_b \end{bmatrix} = \begin{bmatrix} \hat{x}_a & \hat{y}_a & \hat{z}_a \end{bmatrix} {}_b^a R$$

- 따라서 아래와 같이 특정 벡터의 기준좌표계를 회전 행렬을 통해 변환할 수 있다.

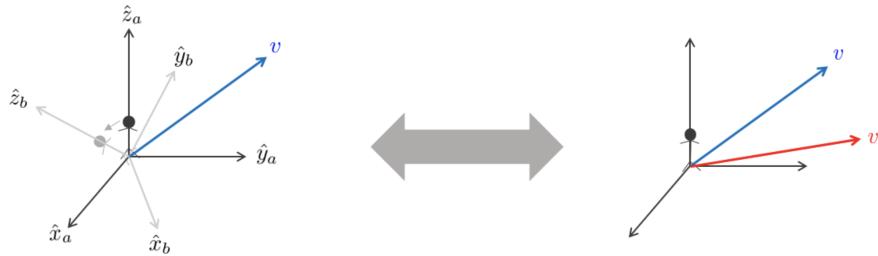
$$\begin{bmatrix} \hat{x}_a & \hat{y}_a & \hat{z}_a \end{bmatrix} \begin{bmatrix} v_{ax} \\ v_{ay} \\ v_{az} \end{bmatrix} = \begin{bmatrix} \hat{x}_a & \hat{y}_a & \hat{z}_a \end{bmatrix} {}_b^a R \begin{bmatrix} v_{bx} \\ v_{by} \\ v_{bz} \end{bmatrix}$$

$$\vec{v}_a = {}_b^a \mathbf{R} \vec{v}_b \quad (15)$$

8. 회전 행렬을 사용하여 특정 벡터를 회전시킬 수도 있다.

$$\mathbf{v}' = {}_b^a \mathbf{R} \mathbf{v} \quad (16)$$

- 좌표계가 회전할 때 좌표계 위에 있는 사람이 보면 벡터가 회전하는 것처럼 보인다.



### 3.2 SO(3) group

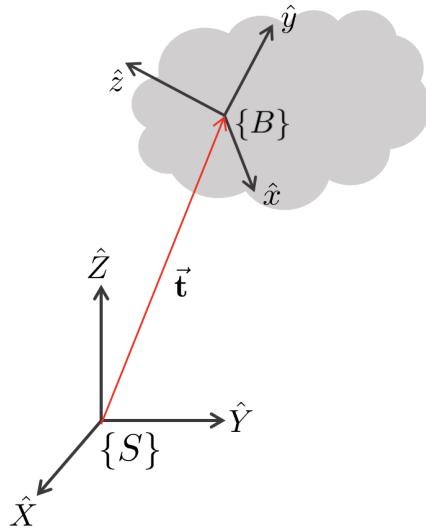
회전 행렬과 같이 순수한 회전에만 관여하는 행렬을 따로 군(group)으로 묶을 수 있다. Lie 군 중 하나인 Special Orthogonal 3(SO(3))군은 3차원 회전 행렬(rotation matrix)과 이에 닫혀있는 연산들로 구성된 군을 의미하며 3차원 물체의 회전을 표현할 때 사용 한다.

$$SO(3) = \{\mathbf{R} \in \mathbb{R}^{3 \times 3} \mid \mathbf{R}\mathbf{R}^T = \mathbf{I}, \det(\mathbf{R}) = 1\} \quad (17)$$

SO(3)군에 대한 자세한 내용은 Lie Theory 개념 정리 (SO(3), SE(3)) 포스트를 참조하면 된다.

## 4 Transformation matrix

이전 섹션에서는 회전 행렬  $\mathbf{R}$ 을 통한 강체의 순수한 회전에 대해 다뤘다. 하지만 3차원 공간 상의 물체는 회전  $\mathbf{R}$ 과 이동  $\mathbf{t}$ 을 모두 표현할 수 있어야 하므로 회전 행렬만으로는 부족하다. 따라서 해당 섹션에서는 물체의 회전과 이동을 한 번에 표현할 수 있는 변환 행렬 (transformation matrix)  $\mathbf{T}$ 에 대해 설명한다.



다시 위 그림으로 돌아와서 고정좌표계  $\{S\}$ 를 기준으로 바위에 붙어있는 이동좌표계  $\{B\}$ 를 표현해보자. 지금까지 정의한 식으로 나타내면 (4)과 같다.

$$\vec{t} = t^\top \begin{bmatrix} \hat{X} \\ \hat{Y} \\ \hat{Z} \end{bmatrix} \quad \begin{bmatrix} \hat{x} \\ \hat{y} \\ \hat{z} \end{bmatrix} = \mathbf{R} \begin{bmatrix} \hat{X} \\ \hat{Y} \\ \hat{Z} \end{bmatrix} \quad (18)$$

- $\mathbf{t} = [t_1, t_2, t_3]^\top$
- $\mathbf{R} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}$

위 식은 원점에 대한 표현  $\vec{t}$ 와 좌표계 회전에 대한 표현  $[\hat{x}, \hat{y}, \hat{z}]^\top$ 을 분리해서 나타내고 있다. 변환 행렬  $\mathbf{T}$ 를 사용하면 다음과 같이 이동과 회전을 한 번에 표현할 수 있다.

$$\mathbf{T} = \left[ \begin{array}{ccc|c} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \\ \hline 0 & 0 & 0 & 1 \end{array} \right] = \left[ \begin{array}{cc} \mathbf{R} & \mathbf{t} \\ 0 & 1 \end{array} \right] \in \mathbb{R}^{4 \times 4} \quad (19)$$

$$\begin{bmatrix} \hat{x} \\ \hat{y} \\ \hat{z} \\ 1 \end{bmatrix} = \mathbf{T} \begin{bmatrix} \hat{X} \\ \hat{Y} \\ \hat{Z} \\ 1 \end{bmatrix} \quad (20)$$

이 때, 변환 행렬  $\mathbf{T}$ 의 마지막 행은 항상  $[0 \ 0 \ 0 \ 1]$ 인데 이는 3차원 벡터와 점을 동차 좌표계(homogeneous coordinate)로 표현했기 때문이다. 따라서 변환 행렬은 동차 변환 행렬(homogeneous transformation matrix)라고도 불린다. 동차 좌표계에 대한 자세한 내용은 Multiple View Geometry in Computer Vision 책 내용 정리 포스팅을 참고하면 된다.

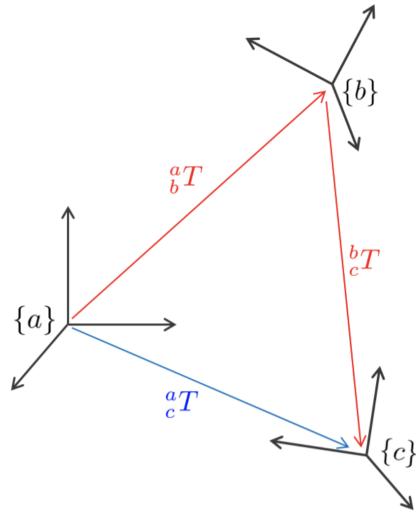
#### 4.1 Properties of transformation matrix

회전 행렬과 동일하게 변환 행렬  $\mathbf{T}$  또한  $\{B\}$  좌표계를 표현하려면 다음과 같이 12개의 파라미터가 필요하다.

$$(t_1, t_2, t_3), (r_{11}, r_{12}, r_{13}, r_{21}, r_{22}, r_{23}, r_{31}, r_{32}, r_{33}) \quad (21)$$

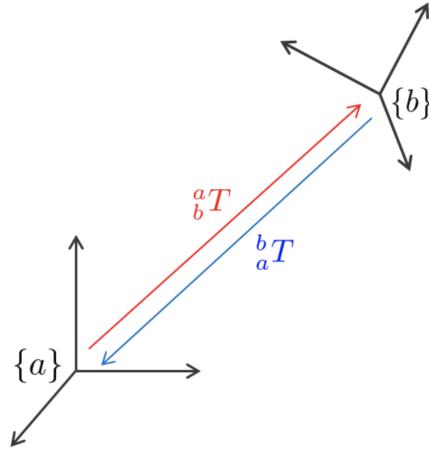
하지만 3차원 공간에서 물체는 회전 3자유도와 이동 3자유도를 합한 6자유도(=6dof, 6 degree of Freedom)를 갖는다. 그리고 회전은 3자유도를 가지므로 이에 따라 12개의 변수는 모두 독립이 아니고 종속적이라는 것을 알 수 있다. 변환 행렬은 회전 행렬을 포함하므로 회전 행렬의 제약 조건을 모두 가진다. 변환 행렬의 성질은 다음과 같다.

1. 연속적인 변환을 나타낼 때는 변환 행렬의 곱셈을 통해 표현할 수 있다.



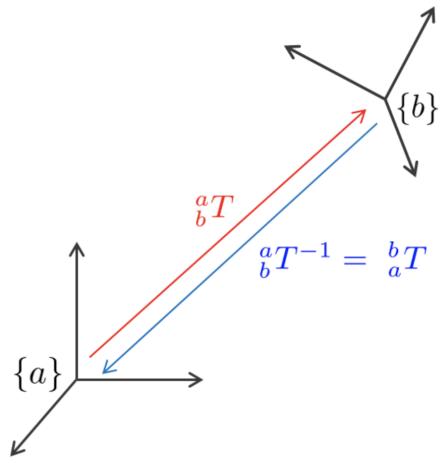
$${}^b_b \mathbf{T} {}^b_c \mathbf{T} = {}^a_c \mathbf{T} \quad \text{where, } {}^a_c \mathbf{T} = ({}^a_c \mathbf{R}, {}^a_c \mathbf{t}) = \begin{bmatrix} {}^a_c \mathbf{R} & {}^a_c \mathbf{t} \\ 0 & 1 \end{bmatrix} \quad (22)$$

2.  $\{a\} \rightarrow \{b\}$ 로 변환한 다음 다시  $\{b\} \rightarrow \{a\}$ 로 변환하면 원래  $\{a\}$  좌표계가 된다. 즉, 변환하지 않은 것과 동일하다.



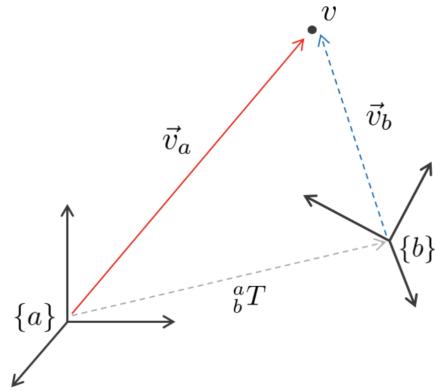
$${}^a_b \mathbf{T} {}^b_a \mathbf{T} = \mathbf{I} \quad (23)$$

3. 변환 행렬의 역행렬은 다음과 같다. 회전 행렬은  $\mathbf{R}^{-1} = \mathbf{R}^T$  이지만 변환 행렬은  $\mathbf{T}^{-1} \neq \mathbf{T}^T$  이다.



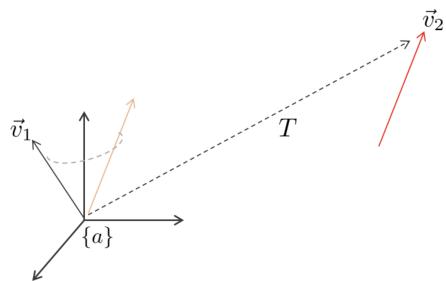
$$\begin{aligned} {}_b^a \mathbf{T} &= \begin{bmatrix} {}_b^a \mathbf{R} & {}_b^a \mathbf{t} \\ 0 & 1 \end{bmatrix} \\ {}_b^a \mathbf{T}^{-1} &= {}_a^b \mathbf{T} = \begin{bmatrix} {}_a^b \mathbf{R}^T & -{}_a^b \mathbf{R}^T {}_b^a \mathbf{t} \\ 0 & 1 \end{bmatrix} \end{aligned} \quad (24)$$

4.  $\{b\}$  좌표계에서 3차원 공간 상의 점  $\mathbf{v}$  를 표현하면  $\vec{v}_b$  와 같고 이를  $\{a\}$  좌표계에서  $\vec{v}_a$  로 표현하기 위해서는 아래와 같은 변환이 필요하다.



$$\begin{aligned} \vec{v}_a &= {}_b^a \mathbf{R} \vec{v}_b + {}_b^a \mathbf{t} \\ \vec{v}_a &= {}_b^a \mathbf{T} \vec{v}_b \end{aligned} \quad (25)$$

5. 좌표계 뿐만 아니라 특정 벡터를 회전 및 이동하는 연산자로써 변환 행렬을 사용할 수 있다. 두  $\{a\}$  좌표계 위의 벡터  $\vec{v}_1, \vec{v}_2$  의 변환관계는 다음과 같이 표현할 수 있다.



$$\begin{bmatrix} x_{v_2} \\ y_{v_2} \\ z_{v_2} \\ 1 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{13} & r_{13} & p_x \\ r_{21} & r_{22} & r_{23} & p_y \\ r_{31} & r_{32} & r_{33} & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{v_1} \\ y_{v_1} \\ z_{v_1} \\ 1 \end{bmatrix}$$

$$\vec{v}_2 = \mathbf{T} \vec{v}_1$$
(26)

## 4.2 SE(3) group

변환 행렬과 같이 물체의 회전과 이동에 관여하는 행렬을 따로 군(group)으로 둘 수 있다. Lie 군 중 하나인 Special Euclidean 3(SE(3))군은 3차원 변환 행렬(transformation matrix)과 이에 닫혀있는 연산들로 구성된 군을 의미하며 3차원 물체의 변환을 표현할 때 사용 한다.

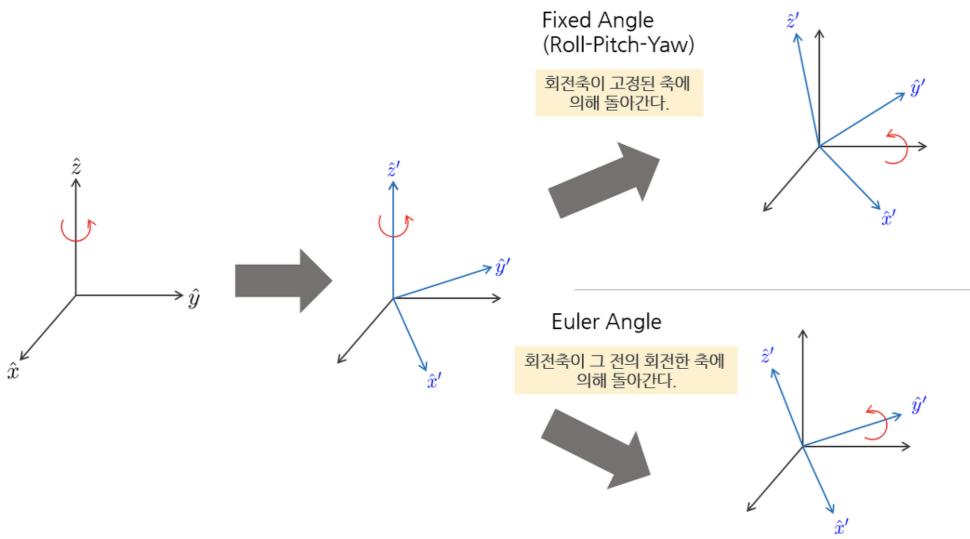
$$SE(3) = \left\{ \mathbf{T} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix} \in \mathbb{R}^{4 \times 4} \mid \mathbf{R} \in SO(3), \mathbf{t} \in \mathbb{R}^3 \right\}$$
(27)

SE(3)군에 대한 자세한 내용은 Lie Theory 개념 정리 (SO(3), SE(3)) 포스트를 참조하면 된다.

## 5 Other rotation representations

### 5.1 Euler angle

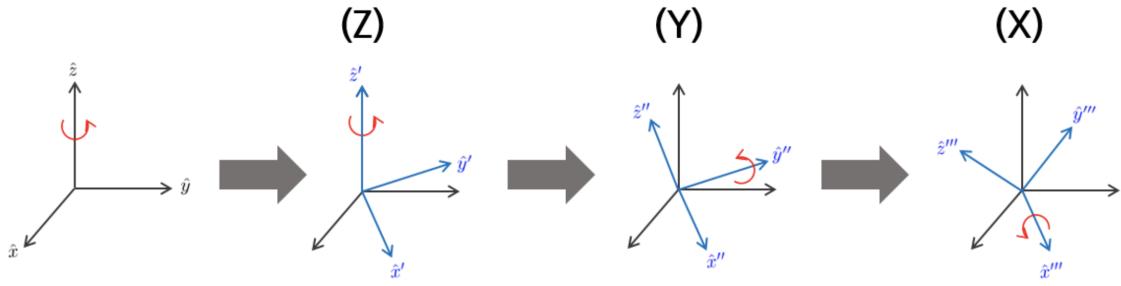
오일러 각도(euler angle) 표현법은 3차원 물체의 방향(orientation)을 3개의 서로 직교하는  $X, Y, Z$ 축 각도로 표현하는 방법 중 하나이다. **3개의 각도로 물체의 방향을 표현할 때 크게 고정 각도(fixed angle) 표현법과 오일러 각도(euler angle) 표현법이 있는데 오일러 각도는 회전한 축을 기준으로 다음 회전을 한다는 점이 고정 각도 표현법과 다르다.** 아래 그림과 같이  $z, y, x$ 축을 순차적으로 돌리는 경우를 생각해보자. 오일러 각도는 회전한 축을 기준으로 다음 회전이 적용된다.



예를 들어, 오일러 각도가  $x, y, z$  축 순서로  $(\gamma, \beta, \alpha)$  라고 하면 이는 우선  $x$  축으로  $\gamma$  만큼 회전한 다음, 회전한 축  $(x', y', z')$  의  $y'$  축을 기준으로  $\beta$  만큼 회전한다. 그리고 마지막으로 회전한 축  $(x'', y'', z'')$  의  $z''$  축을 기준으로  $\alpha$  만큼 회전한다.

#### 5.1.1 Various euler angle representations

오일러 각도는 회전 순서에 따라 XYZ, ZXY, ZXZ, XYX, YXY, YZY, ZYZ, ZXZ, XZY, YXZ, YZX, ZYX와 같이 12개의 서로 다른 표현법이 존재한다. 각 회전은 이전 회전에 종속적이며 교환법칙이 성립하지 않으므로 축을 돌리는 순서를 아는 것이 중요하다. 예를 들어 ZYX의 경우 아래 그림과 같다.

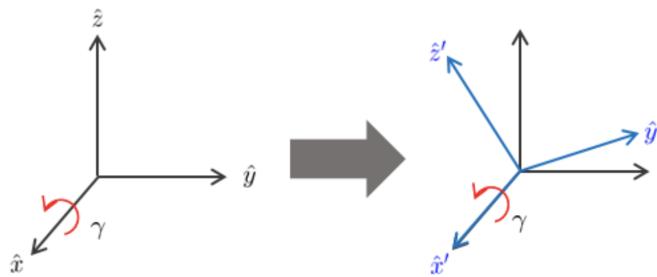


이 때, 오일러 각도는 다음과 모든 회전을 같이 하나로 합쳐서 표현한다.

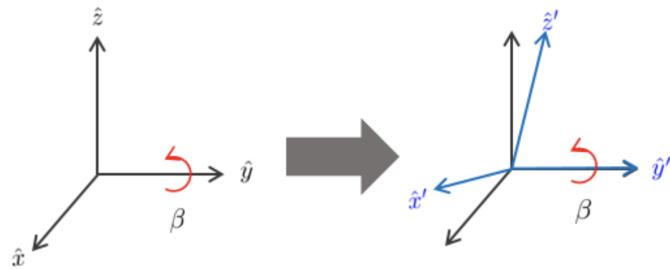
$$\begin{aligned}
 R_{ZYX}(\alpha, \beta, \gamma) &= \text{Rot}(\hat{z}, \alpha) \text{Rot}(\hat{y}, \beta) \text{Rot}(\hat{x}, \gamma) = R_Z(\alpha) R_Y(\beta) R_X(\gamma) \\
 &= \begin{bmatrix} c_\alpha c_\beta & c_\alpha s_\beta s_\gamma - s_\alpha c_\gamma & c_\alpha s_\beta c_\gamma + s_\alpha s_\gamma \\ s_\alpha c_\beta & s_\alpha s_\beta s_\gamma + c_\alpha c_\gamma & s_\alpha s_\beta c_\gamma - c_\alpha s_\gamma \\ -s_\beta & c_\beta s_\gamma & c_\beta c_\gamma \end{bmatrix} \quad (28)
 \end{aligned}$$

### 5.1.2 Relationship between euler angle and rotation matrix

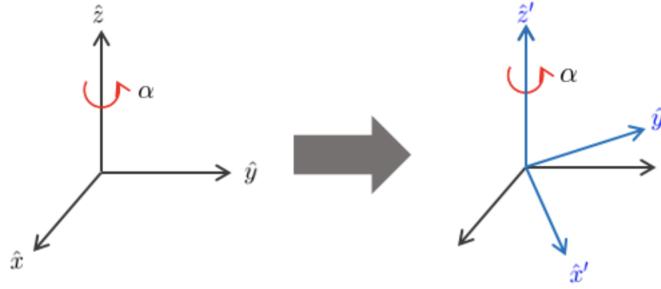
모든 오일러 각도는 회전 행렬로 변환 가능하다. 아래 그림과 같이  $\hat{x}$ ,  $\hat{y}$ ,  $\hat{z}$  축을 기준으로 좌표계를 각각  $\gamma, \beta, \alpha$  만큼 회전시킬 때 회전 행렬은 다음과 같다.



$$R_x(\gamma) = \text{Rot}(\hat{x}, \gamma) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \gamma & -\sin \gamma \\ 0 & \sin \gamma & \cos \gamma \end{bmatrix} \quad (29)$$



$$R_y(\beta) = \text{Rot}(\hat{y}, \beta) = \begin{bmatrix} \cos \beta & 0 & \sin \beta \\ 0 & 1 & 0 \\ -\sin \beta & 0 & \cos \beta \end{bmatrix} \quad (30)$$



$$R_z(\alpha) = \text{Rot}(\hat{z}, \alpha) = \begin{bmatrix} \cos \alpha & -\sin \alpha & 0 \\ \sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (31)$$

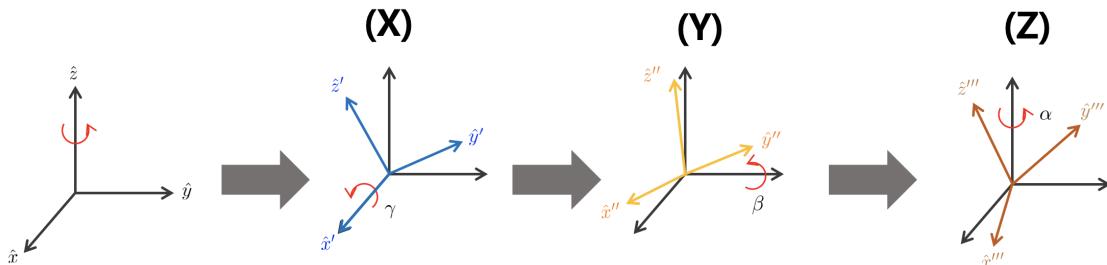
### 5.1.3 Limitation of euler angle

오일러 각도는 3차원 상의 물체의 방향(orientation)을 3개의 파라미터만으로 간결하게 표현할 수 있다는 점과 대략적인 방향에 대한 직관적인 정보를 준다는 점에서 좋지만 다음과 같은 한계점을 가지고 있다.

1. **Gimbal Lock:** 오일러 각도 표현 법은 짐벌락이라는 치명적인 문제점을 가지고 있다. 오일러 각도로 물체를 표현할 때 같은 방향으로 두 회전 축이 겹치는 경우 해당 2개의 축이 더이상 분리되지 않는 고정 상태가 되는데 이를 짐벌락이라고 한다. 따라서 3개의 자유도 중 1개의 자유도를 잊어버리게 되므로 다양한 방향 표현이 불가능해진다. 짐벌락 유튜브
2. **Interpolation:** 오일러 각도 표현법은 고정 각도 표현법과 다르게 3축에 대해 회전이 종속적이므로 2개의 오일러 각도 사이를 보간(interpolation)할 때에도 문제가 발생한다. 이는 컴퓨터로 구현할 때 문제가 된다. 따라서 물체의 방향(orientation)을 표현할 때는 오일러 각도를 종종 사용하지만 물체의 회전(rotation)을 표현할 때는 오일러 각도를 거의 사용하지 않는다.
  - 예를 들어  $R_{ZYX}(0, 180, 0) = R_{ZYX}(180, 0, 180)$  은 같은 물체의 방향(orientation)을 표현한다. 이 때  $(0, 0, 0) \rightarrow (0, 180, 0)$  으로 보간하는 경우 중간값은  $(0, 90, 0)$  이지만  $(0, 0, 0) \rightarrow (180, 0, 180)$  으로 보간하는 경우 중간값은  $(90, 0, 90)$  이 되어 서로 완전 다른 방향으로 회전하게 된다.

## 5.2 Fixed angle (roll-pitch-yaw)

고정 각도(fixed angle) 표현법은 3차원 물체의 방향(orientation)을 3개의 서로 직교하는  $x, y, z$  축 각도로 표현하는 방법 중 하나이다. 3개의 각도로 물체의 방향을 표현할 때 크게 고정 각도(fixed angle) 표현법과 오일러 각도(euler angle) 표현법이 있는데 고정 각도는 고정된 축을 기준으로 다음 회전을 한다는 점이 오일러 각도 표현법과 다르다.



위와 같이  $x, y, z$  축 순서대로 회전하는 경우 새로운 축은 아래 공식과 같이 나타낼 수 있다.

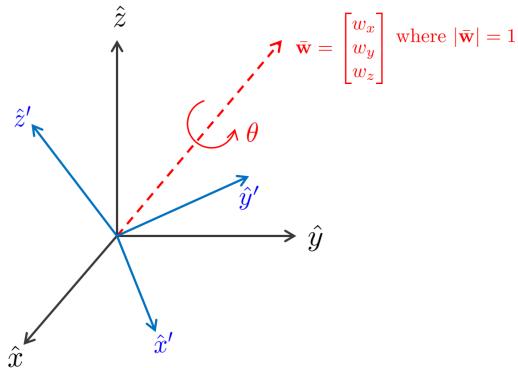
$$\begin{aligned} [\hat{x}' \ \hat{y}' \ \hat{z}'] &= \text{Rot}(\hat{X}, \gamma) [\hat{x} \ \hat{y} \ \hat{z}] \\ [\hat{x}'' \ \hat{y}'' \ \hat{z}''] &= \text{Rot}(\hat{Y}, \beta) [\hat{x}' \ \hat{y}' \ \hat{z}'] \\ [\hat{x}''' \ \hat{y}''' \ \hat{z}'''] &= \text{Rot}(\hat{Z}, \alpha) [\hat{x}'' \ \hat{y}'' \ \hat{z}''] \end{aligned}$$

이를 한번에 정리하면 아래와 같다. 이 때  $\alpha, \beta, \gamma$ 를 각각 yaw, pitch, roll이라고 한다.

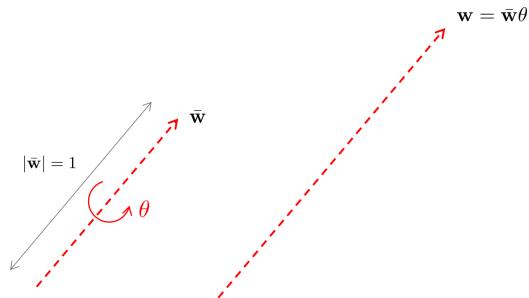
$$[\hat{x}''' \quad \hat{y}''' \quad \hat{z}'''] = \text{Rot}(\hat{Z}, \alpha) \text{Rot}(\hat{Y}, \beta) \text{Rot}(\hat{X}, \gamma) [\hat{x} \quad \hat{y} \quad \hat{z}] \quad (32)$$

위 식은 (28)식과 동일하다. 따라서 고정 각도 표현법의 XYZ 회전은 오일러 각도 표현법의 ZXY와 동일한 회전을 의미한다.

### 5.3 Angle-axis representation



Angle-axis 표현법은 회전을 하나의 축(axis)과 이를 기준으로 회전한 각도(angle)로 표현하는 방법을 말한다. Angle-axis 표현법은 회전 축  $[w_x \ w_y \ w_z]$  3개와 회전 각도  $\theta$  1개의 파라미터를 사용하여 총 4개의 파라미터로 회전을 표현할 수 있다.



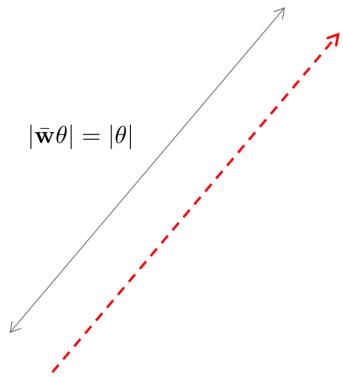
4개의 파라미터를 합치면 하나의 벡터로 표현할 수 있다. 이는 회전 벡터(rotation vector), rodrigues 파라미터, 각속도(angular velocity) 벡터 등으로 불린다.

$$\mathbf{w} = \bar{\mathbf{w}}\theta = \begin{bmatrix} w_x\theta \\ w_y\theta \\ w_z\theta \end{bmatrix} \quad (33)$$

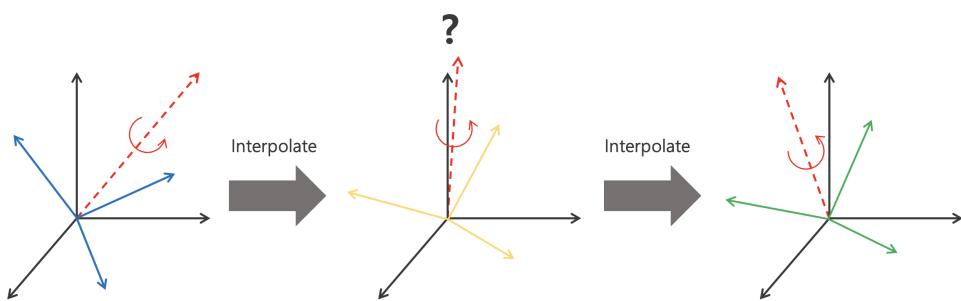
### 5.4 Properties of angle-axis representation

- Redundancy:** 3차원 물체의 회전을 표현하기 위한 최소한의 파라미터는 3개이지만 angle-axis 표현법은 파라미터가 4개가 필요하다. 이는 3dof를 4dof로 표현한 셈이 되므로 angle-axis 표현법은 redundancy를 갖는다. 이러한 redundancy는 회전 벡터의 크기를 의미한다. 이 때, 회전 벡터  $\bar{\mathbf{w}}$ 의 크기  $|\bar{\mathbf{w}}|$ 를 정규화(normalize)하게 되면  $\bar{\mathbf{w}}$ 는 1dof를 잃게 되고 따라서 redundancy를 제거할 수 있다.

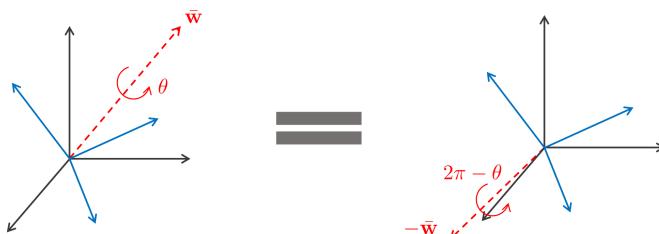
$$|\bar{\mathbf{w}}| = 1 \quad (34)$$



2. 오일러 각도 표현법에서 발생한 gimbal lock 문제가 발생하지 않는다.
3. 두 개의 회전 사이를 보간하는 경우 회전이 부드럽지 못한 문제가 존재한다.



4. 하나에 회전에 대하여 회전 벡터가 유일하지 않는 문제를 가지고 있다.



## 5.5 Quaternion

쿼터니언(quaternion)은 4개의 값으로 이루어진 복소수 체계이다. 쿼터니언 자체는 회전과 관련없는 독립적인 복소수 체계이지만 이 중 단위 쿼터니언(unit quaternion)이 3차원 공간 상의 물체의 회전을 표현하는 연산자로 사용된다. 이는 컴퓨터 3D 그래픽스 분야에서 광범위하게 사용되며 실제로 회전 표현에 있어서 가장 효과적인 방법이다. 쿼터니언을 사용하면 다양한 회전 표현들의 문제점들을 회피할 수 있다.

1. gimbal lock 현상이 발생하지 않는다.
2. 행렬보다 점유 메모리 영역이 작고 계산부하도 낮다.
3. 임의의 회전 벡터(angle-axis)를 손쉽게 쿼터니언으로 변환할 수 있다.
4. 회전을 간단하게 합성할 수 있고 오차도 거의 발생하지 않는다.
5. 두 개의 회전 사이의 보간을 매끄럽게 수행할 수 있다.

하지만 4차원 복소수이므로 시각화가 되지 않아 직관적으로 이해하기 어렵다는 단점이 존재한다. 쿼터니언은 다음과 같이 실수  $w$ 와 허수  $i, j, k$ 와 대응하는  $x, y, z$  값으로 구성되어 있다.

$$\begin{aligned} \mathbf{q} &= [w \ x \ y \ z]^\top \\ &= w + xi + yj + zk \end{aligned} \tag{35}$$

위와 같이  $[w \ x \ y \ z]$  순서로 표현하고  $ijk = -1$ 을 만족하는 오른손 법칙을 따르는 쿼터니언 표기법을 'hamilton' 쿼터니언이라고도 부른다. 이와 반대로  $[x \ y \ z \ w]$ 로 표현하고 왼손 법칙을 따르는 쿼터니언을 'JPL' 쿼터니언이라고도 부른다. 이에 대한 자세한 내용은 Quaternion kinematics for the error-state Kalman filter 내용 정리 포스팅을 참고하면 된다. 본 섹션에서는 hamilton 쿼터니언 표기법으로 설명한다.

### 5.5.1 Properties of quaternion

1. **Addition:** 두 쿼터니언의 덧셈은 실수부와 허수부를 나눠서 수행한다.

$$\mathbf{p} + \mathbf{q} = \begin{bmatrix} p_w \\ \mathbf{p}_v \end{bmatrix} \pm \begin{bmatrix} q_w \\ \mathbf{q}_v \end{bmatrix} = \begin{bmatrix} p_w \pm q_w \\ \mathbf{p}_v \pm \mathbf{q}_v \end{bmatrix} \quad (36)$$

2. **Multiplication:** 두 hamilton 쿼터니언의 곱은  $\otimes$ 로 표기하며 아래와 같다.

$$\mathbf{p} \otimes \mathbf{q} = \begin{bmatrix} p_w q_w - p_x q_x - p_y q_y - p_z q_z \\ p_w q_x + p_x q_w + p_y q_z - p_z q_y \\ p_w q_y - p_x q_z + p_y q_w + p_z q_x \\ p_w q_z + p_x q_y - p_y q_x + p_z q_w \end{bmatrix} \quad (37)$$

3. **Identity:** 항등 쿼터니언(identity quaternion)  $\mathbf{q}_1$ 은 다음과 같은 공식  $\mathbf{q}_1 \otimes \mathbf{q} = \mathbf{q} \otimes \mathbf{q}_1 = \mathbf{q}$ 를 만족하며 다음과 같이 정의된다.

$$\mathbf{q}_1 = 1 = \begin{bmatrix} 1 \\ \mathbf{0}_v \end{bmatrix} \quad (38)$$

4. **Conjugate:** 쿼터니언의 결레복소수는 다음과 같이 정의된다.

$$\mathbf{q}^* \triangleq q_w - \mathbf{q}_v = \begin{bmatrix} q_w \\ -\mathbf{q}_v \end{bmatrix} \quad (39)$$

5. **Norm:** 쿼터니언의 놈은 다음과 같이 정의된다.

$$\|\mathbf{q}\| \triangleq \sqrt{\mathbf{q} \otimes \mathbf{q}^*} = \sqrt{\mathbf{q}^* \otimes \mathbf{q}} = \sqrt{q_w^2 + q_x^2 + q_y^2 + q_z^2} \in \mathbb{R} \quad (40)$$

6. **Inverse:** 쿼터니언의 역은  $\mathbf{q}^{-1}$ 과 같이 표기하며 다음과 같은 특성을 지닌다.

$$\mathbf{q} \otimes \mathbf{q}^{-1} = \mathbf{q}^{-1} \otimes \mathbf{q} = \mathbf{q}_1 \quad (41)$$

7. **Unit quaternion:** 단위 쿼터니언은  $\|\mathbf{q}\| = 1$ 의 성질을 지닌다. 따라서 아래와 같은 특성을 만족한다.

$$\mathbf{q}^{-1} = \mathbf{q}^* \quad (42)$$

**단위 쿼터니언은 3차원 공간에서 물체의 방향을 표현하거나 회전을 수행하는 연산자로써 사용된다.** 위 성질은 역방향 회전이 기준 벡터에 결레복소수 쿼터니언을 곱함으로써 얻어질 수 있다는 것을 의미한다. 단위 쿼터니언은 항상 다음과 같이 표현될 수 있다.

$$\mathbf{q} = \begin{bmatrix} \cos \theta \\ \mathbf{u} \sin \theta \end{bmatrix} \quad (43)$$

이 때,  $\mathbf{u} = u_x i + u_y j + u_z k$ 는 단위 벡터이며  $\theta$ 는 스칼라 값이다.  $\mathbf{u}, \theta$ 는 angle-axis로 표현한 회전 벡터 파라미터를 의미하므로 angle-axis 표현법과 쿼터니언이 쉽게 상호 변환할 수 있음을 알 수 있다.

8. **Pure quaternion:** 실수부  $w = 0$ 이고 허수부는 0이 아닐 때 이를 순수 쿼터니언(pure quaternion)이라고 한다.

$$\mathbf{q} = [0 \ x \ y \ z]^\top \quad (44)$$

### 5.5.2 Unit quaternion as a rotation operator

단위 쿼터니언이 임의의 3차원 공간 상의 벡터  $\mathbf{x}$ 를 회전시키려면 다음과 같이 연산이 수행되어야 한다.

$$\boxed{\mathbf{x}' = \mathbf{q} \otimes \mathbf{x} \otimes \mathbf{q}^*} \quad (45)$$

이 때,  $\mathbf{x}'$ 는 순수 쿼터니언(pure quaternion) 형태로 허수부에 3차원 벡터의 원소가 들어간다. 실수부  $w$ 는 0으로 설정한다. 쿼터니언 회전을 염밀하게 표현하면 다음과 같다.

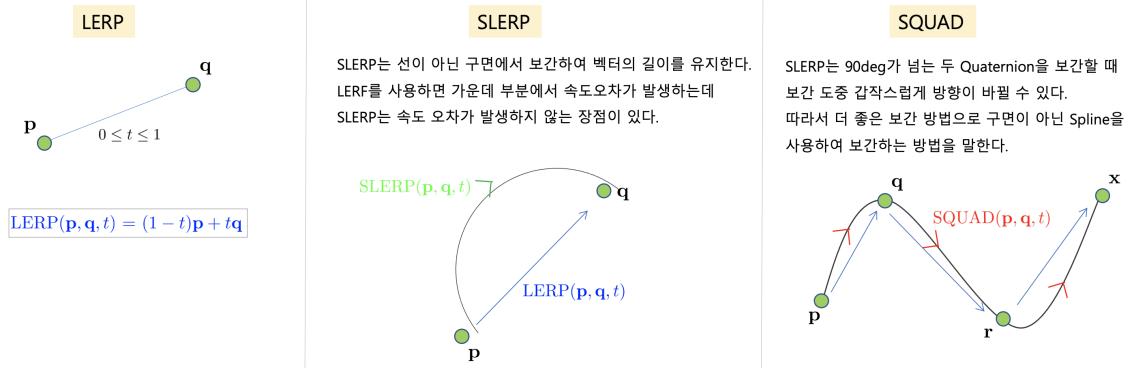
$$\begin{bmatrix} 0 \\ \mathbf{x}' \end{bmatrix} = \mathbf{q} \otimes \begin{bmatrix} 0 \\ \mathbf{x} \end{bmatrix} \otimes \mathbf{q}^* \quad (46)$$

### 5.5.3 Interpolation

쿼터니언은 두 회전 사이의 보간(interpolation)을 가장 효과적으로 표현할 수 있다. 보간 방법의 종류는 다음과 같다.

- Linear Interpolation (LERP)
- Spherical Linear Interpolation (SLERP)
- Spherical Cubic Interpolation (SQUAD)

이를 비교한 그림은 다음과 같다. 쿼터니언 보간에 대해 보다 자세히 알고 싶으면 위키피디아를 참조하면 된다.



## 6 Relationship between rotation representations

지금까지 설명한 회전 행렬(9), 오일러 각도(3), 고정 각도(3), angle-axis(4), 쿼터니언(4) 표현 방법은 모두 3차원 공간 상의 회전을 표현하는 방법이다. 이러한 회전 표현법 사이에는 서로 변환이 가능하다.

Rotation Matrix(9)	Euler Angle(3)	Fixed Angle(3)	Angle-Axis(4)	Quaternion(4)
$\mathbf{R} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}$	Euler $(\alpha, \beta, \gamma)$	RPY $(\alpha, \beta, \gamma)$	$\mathbf{w} = \bar{\mathbf{w}}\theta = \begin{bmatrix} w_x\theta \\ w_y\theta \\ w_z\theta \end{bmatrix}$	$\mathbf{q} = [w \ x \ y \ z]$

### 6.1 Angle-axis and quaternion

Angle-axis  $\mathbf{w} = \bar{\mathbf{w}}\theta = \begin{bmatrix} w_x\theta \\ w_y\theta \\ w_z\theta \end{bmatrix}$  가 주어졌을 때 쿼터니언은 다음과 같이 구할 수 있다.

$$\mathbf{w} = \bar{\mathbf{w}}\theta = \begin{bmatrix} w_x\theta \\ w_y\theta \\ w_z\theta \end{bmatrix} \rightarrow \mathbf{q} = \begin{bmatrix} \cos \frac{\theta}{2} \\ \bar{\mathbf{w}} \sin \frac{\theta}{2} \end{bmatrix}$$

(47)

이 때, 회전 축  $\bar{\mathbf{w}}$ 가 단위 벡터이면 쿼터니언도 단위 쿼터니언이 된다. 이와 반대로 쿼터니언  $\mathbf{q} = [w, x, y, z]^\top$ 이 주어졌을 때 angle-axis는 다음과 같이 구할 수 있다.  $w$ 는 쿼터니언의 실수부이고  $w_x, w_y, w_z$ 는 angle-axis의 단위 벡터임에 유의한다.

$$\mathbf{q} = \begin{bmatrix} w \\ x \\ y \\ z \end{bmatrix} \rightarrow \begin{pmatrix} s = \sqrt{x^2 + y^2 + z^2} \\ w_x = x/s \\ w_y = y/s \\ w_z = z/s \\ \theta = 2 \cos^{-1}(w) \end{pmatrix}$$

(48)

## 6.2 Rotation matrix and quaternion

회전 행렬  $\mathbf{R} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}$  이 주어져 있을 때 쿼터니언은 다음과 같이 구할 수 있다.

$$\mathbf{R} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \rightarrow \mathbf{q} = \begin{bmatrix} w = \frac{\sqrt{r_{11}+r_{22}+r_{33}+1}}{2} \\ x = \frac{r_{23}-r_{32}}{4w} \\ y = \frac{r_{31}-r_{13}}{4w} \\ z = \frac{r_{12}-r_{21}}{4w} \end{bmatrix} \quad (49)$$

만약  $w = 0$ 인 경우 위 식이 성립하지 않는다. 이런 경우에는  $w, x, y, z$  중 가장 큰 값을 찾고 이를 통해 나머지 세 개의 값을 계산해야 한다. 이와 반대로 쿼터니언  $\mathbf{q} = [w, x, y, z]^\top$ 이 주어졌을 때 회전 행렬은 다음과 같이 구할 수 있다.

$$\mathbf{q} = [w, x, y, z]^\top \rightarrow \mathbf{R} = \begin{bmatrix} x^2 + y_2 - z^2 + w^2 & 2xy - 2wz & 2xz + 2wy \\ 2xy + 2wz & -x^2 + y_2 - z^2 + w^2 & 2yz - 2wx \\ 2xz - 2wy & 2yz + 2wx & -x^2 - y_2 + z^2 + w^2 \end{bmatrix} \quad (50)$$

## 6.3 Rotation matrix and angle-axis

회전 행렬  $\mathbf{R} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}$  이 주어져 있을 때 angle-axis은 다음과 같이 구할 수 있다.

$$\mathbf{R} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \rightarrow \begin{pmatrix} \theta = \arccos\left(\frac{r_{11}+r_{22}+r_{33}-1}{2}\right) \\ \bar{\mathbf{w}} = \frac{1}{2\sin\theta} \begin{bmatrix} r_{32}-r_{23} \\ r_{13}-r_{31} \\ r_{21}-r_{12} \end{bmatrix} \end{pmatrix} \quad (51)$$

이와 반대로 angle-axis  $\mathbf{w} = \bar{\mathbf{w}}\theta = \begin{bmatrix} w_x\theta \\ w_y\theta \\ w_z\theta \end{bmatrix}$  가 주어졌을 때 회전 행렬은 다음과 같이 구할 수 있다.

$$\mathbf{w} = \bar{\mathbf{w}}\theta = \begin{bmatrix} w_x\theta \\ w_y\theta \\ w_z\theta \end{bmatrix} \rightarrow \mathbf{R} = \begin{bmatrix} w_x^2 v\theta + c\theta & w_x w_y v\theta - w_z s\theta & w_x w_z v\theta + w_y s\theta \\ w_x w_y v\theta + w_z s\theta & w_y^2 v\theta + c\theta & w_y w_z v\theta - w_x s\theta \\ w_x w_z v\theta - w_y s\theta & w_y w_z v\theta + w_x s\theta & w_z^2 v\theta + c\theta \end{bmatrix} \quad (52)$$

- $s\theta = \sin\theta$
- $c\theta = \cos\theta$
- $v\theta = 1 - \cos\theta$

위 식을 컴팩트하게 표현하면 아래와 같이 나타낼 수 있는데 이를 Rodrigues 공식이라고 한다.

$$\mathbf{w} = \bar{\mathbf{w}}\theta = \begin{bmatrix} w_x\theta \\ w_y\theta \\ w_z\theta \end{bmatrix} \rightarrow \mathbf{R} = \mathbf{I} + \sin\theta[\bar{\mathbf{w}}]_\times + (1 - \cos\theta)[\bar{\mathbf{w}}]^2_\times \quad (53)$$

## 6.4 Quaternion and euler angle(ZYX)

쿼터니언  $\mathbf{q} = [w, x, y, z]^\top$ 이 주어졌을 때 오일러 각도는 다음과 같이 구할 수 있다.

$$\mathbf{q} = [w, x, y, z]^\top \rightarrow \begin{bmatrix} \alpha \\ \beta \\ \gamma \end{bmatrix} = \begin{bmatrix} \arctan \frac{2(wx+yz)}{1-2(x^2+y^2)} \\ \arcsin(2(wy-xz)) \\ \arctan \frac{2(wz+xy)}{1-2(y^2+z^2)} \end{bmatrix} \quad (54)$$

이와 반대로 오일러 각도  $\gamma, \beta, \alpha$ 가 주어져 있을 때 쿼터니언은 다음과 같이 구할 수 있다.

$$\begin{aligned} \text{Euler}_{ZYX}(\alpha, \beta, \gamma) &\rightarrow x = \sin \frac{\beta}{2} \cos \frac{\alpha - \gamma}{2} \\ &y = \sin \frac{\beta}{2} \sin \frac{\alpha - \gamma}{2} \\ &z = \cos \frac{\beta}{2} \sin \frac{\alpha + \gamma}{2} \\ &w = \cos \frac{\beta}{2} \cos \frac{\alpha + \gamma}{2} \end{aligned} \quad (55)$$

## 6.5 Rotation matrix and euler angle(ZYX)

오일러 각도  $\gamma, \beta, \alpha$ 가 주어져 있을 때 회전 행렬은 다음과 같이 구할 수 있다.

$$\text{Euler}_{ZYX}(\alpha, \beta, \gamma) \rightarrow \mathbf{R} = \begin{bmatrix} c_\alpha c_\beta & c_\alpha s_\beta s_\gamma - s_\alpha c_\gamma & c_\alpha s_\beta c_\gamma + s_\alpha s_\gamma \\ s_\alpha c_\beta & s_\alpha s_\beta s_\gamma + c_\alpha c_\gamma & s_\alpha s_\beta c_\gamma - c_\alpha s_\gamma \\ -s_\beta & c_\beta s_\gamma & c_\beta c_\gamma \end{bmatrix} \quad (56)$$

- $s_\alpha = \sin \alpha$
- $c_\alpha = \cos \alpha$

이와 반대로 회전 행렬이 주어졌을 때 오일러 각도를 구하는 공식은 수식이 매우 길고 복잡하며 잘 사용하지 않기 때문에 생략한다. 자세한 내용은 해당 링크를 참조하면 된다.

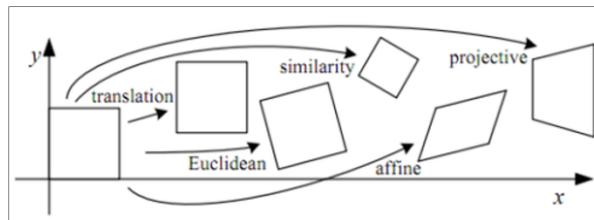
## 7 Reference

1. SLAMKR Study Season 1
2. Robotic Systems Chapter 4. 3D Rotations - Duke university
3. Keunjun's blog

## 8 Revision log

- |                   |                   |                   |                    |
|-------------------|-------------------|-------------------|--------------------|
| • 1st: 2019-07-06 | • 4th: 2019-07-13 | • 7th: 2023-01-24 | • 10th: 2023-11-30 |
| • 2nd: 2019-07-11 | • 5th: 2019-10-30 | • 8th: 2023-01-26 |                    |
| • 3rd: 2019-07-12 | • 6th: 2023-01-23 | • 9th: 2023-01-28 |                    |

## A Various transformations



우선 2차원 평면에서 사각형 물체의 변환에 대해 설명한다. 위 그림과 같이 물체를 변환하는 다양한 연산들이 존재하며 이는 기존 사각형을 구성하는  $n$ 개의 벡터  $\mathbf{X} \in \mathbb{R}^{2 \times n}$ 에 변환 행렬  $\mathbf{H} \in \mathbb{R}^{2 \times 2}$ 를 곱함으로써 적용할 수 있다.

$$\begin{aligned} \mathbf{X}' &= \mathbf{H}\mathbf{X} \\ \mathbb{R}^{2 \times n} &= \mathbb{R}^{2 \times 2} \mathbb{R}^{2 \times n} \end{aligned} \quad (57)$$

## A.1 Isometry transformation

물체가 변환 전과 후 사이에 물체의 크기가 동일한 경우 해당 변환을 isometry 변환이라고 부른다. **Isometry 변환에는 translation, euclidean 변환이 있으며 이 중 eculidean 변환이 곧 강체 변환을 의미한다.**

$$\mathbf{H}_{iso} = \begin{bmatrix} \mathbf{A} & \mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix} \quad (58)$$

이 때  $\mathbf{A} \in \mathbb{R}^{2 \times 2}$ 는 2차원 물체의 회전과 반사(reflection)을 포함한 행렬이며  $\mathbf{t} \in \mathbb{R}^2$ 는 2차원 물체의 이동 벡터이다. Euclidean 변환은 물체를 회전 후 이동시키므로 다음과 같이 나타낼 수 있다.

$$\mathbf{H}_{euc} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix} \quad (59)$$

## A.2 Similarity transformation

Isometry 변환에 스케일을 의미하는  $s$ 가 추가된 변환을 similarity 변환이라고 하며 물체의 이동 및 회전에 추가적으로 스케일까지 같이 변환하는 성질을 가진다. 이 때 기존의  $\mathbf{A}$  행렬에서 reflection의 성질을 제거한  $\mathbf{R}$  행렬을 사용한다.

$$\mathbf{H}_S = \begin{bmatrix} s\mathbf{R} & \mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix} \quad (60)$$

similarity 변환은 물체의 각도와 길이의 비율을 보존되지만 스케일은 보존되지 않는다. 두 물체가 similarity 변환까지 동일하다는 의미는(=up to scale) 두 물체의 형태는 동일하지만 스케일에 차이가 있다는 의미이다.

## A.3 Affine transformation

affine 변환은 isometry 변환에서 행렬  $\mathbf{A}$ 의 아무 제약조건이 없는 변환 행렬을 의미한다. 변환 후 물체는 일반적으로 변환 전과 다른 형태를 가진다.

$$\mathbf{H}_A = \begin{bmatrix} \mathbf{A} & \mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix} \quad (61)$$

## A.4 Projective transformation

마지막으로 projective 변환은 변환 행렬의 마지막 행이  $(0, 0, 1)$ 이 아닌 임의의 형태를 가지는 변환 행렬을 의미한다. projective 변환은 직선을 직선으로 사상한다는 성질 외에는 이전에 설명한 모든 성질들을 보존하지 않는다는 특징이 있다. 평행한 직선 또한 projective 변환을 수행하면 평행하지 않게되며 물체의 길이비 또한 달라지게 된다.

$$\mathbf{H}_P = \begin{bmatrix} \mathbf{A} & \mathbf{t} \\ \mathbf{v}^\top & v \end{bmatrix} \quad (62)$$

이 때  $\mathbf{v} = [v_1 \ v_2]^\top$ 는 임의의 2차원 벡터를 의미하며  $v$  또한 임의의 스칼라 값을 의미한다.